IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Reversible Data Hiding Using Slicer Strategy

[1]Afsiya S, Archana, Shibin S Tom, Vyshak Prakash

[2]Prof.Harsha Pushpan

[1](Final Year Students, Dept of Computer Science & Engineering, Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India)

[2](Assistant Professor, Department of Computer Science & Engineering, Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India)

Abstract: With the popularity of wireless devices equipped with various kinds of sensing abilities, a new service paradigm named participatory sensing has emerged to provide users with brand new life experience. However, the wide application of participatory sensing has its own challenges, among which privacy and multimedia data quality preservations are two critical problems. Unfortunately, none of the existing work has fully solved the problem of privacy and quality preserving participatory sensing with multimedia data. In this paper, we propose SLICER, which is the first k-anonymous privacy preserving scheme for participatory sensing with multimedia data. SLICER integrates a data coding technique and message transfer strategies, to achieve strong protection of participants privacy, while maintaining high data quality with secure password to the application using encrypted negative password. Data is split using slicer, each split file is encrypted using an encryption algorithm and each encrypted file is compressed using a compression algorithm and stored in a server.

***Index Terms* – Slicer Strategy, Compression Algorithm.**

INTRODUCTION

Data hiding and picture slicing are combined in a novel way to provide a highly effective and reversible data hiding strategy known as reversible data hiding using slicer technology. Strong and reversible techniques are now more necessary than ever in the field of data security and information concealment. The goal of reversible data hiding is to include data into a host signal while making sure the original data can be fully recovered without any loss or deterioration.

The slicer technology achieves reversible data concealment by utilising the properties of the host image. It creates many slices or blocks out of the host image, making the processes for embedding and extracting data more effective. As a result of treating each slice as a separate entity, the data concealing method can focus on particular areas of the image without influencing the rest of it. Reversible data concealing with slicer technology has the major benefit of high embedding capacity with the least amount of host image distortion. The program strikes a compromise between data capacity and visual quality by carefully choosing the slices and embedding data within them. This method offers a reversible data concealment mechanism that is strong and unnoticeable to the human sight, enabling the smooth recovery of the original image.

Slicer technology is used for reversible data concealment in a number of fields, such as multimedia transmission, copyright protection, and data authentication. It is a useful tool in situations when data integrity and secrecy are essential since it can embed and extract data without any loss or damage to the host signal. Using slicer technology, reversible data hiding provides a potent method for integrating

data into host images while preserving reversibility and aesthetic quality. This strategy strikes a compromise between embedding capacity and imperceptibility by exploiting picture slicing techniques, making it a viable solution in the realm of data security and information concealment.

The idea of a negative password is distinctive and exciting in the realm of information security since it goes against the grain of traditional ideas about password-based authentication. Negative passwords employ a different strategy than conventional ones by relying on the public exposure of specific information that is used as a form of authentication as opposed to regular passwords, which depend on the secrecy and complexity of the selected string of characters.

Instead of depending only on a secret password, users of negative password systems voluntarily share certain facts or characteristics about themselves that can be easily verified. Personal information, such as biometric data, physical traits, or even publicly available data, like a birth date or address, might be included in these specifics. The exposed knowledge is not supposed to be kept a secret; rather, it is designed to be shared publicly.

Negative passwords are designed so that even while an attacker has access to the same publicly available information, they do not have the same context or relationship with it as a normal user. An attacker, for instance, might know someone's birthdate, but they wouldn't have access to the memories, feelings, or experiences that are personally connected to that date and that the user can use to verify their identity.

Negative password systems strive to develop a more dependable and user-friendly authentication method by harnessing the inherent individuality and uniqueness of human experiences. It minimises the possibility of password reuse across several platforms and lessens the cognitive load of memorising complex passwords.

Negative passwords and reversible data concealment with slicer technology are two different ideas, yet they both have an impact on the whole information security landscape. While reversible data hiding with slicer technology concentrates on safeguarding data within a host signal while assuring reversibility, negative passwords seek to improve authentication procedures by utilising personal context. Both fields are still developing and have bright opportunities for dealing with security issues in all facets of digital systems.

LITERATURE SURVEY

1.Reversible data hiding method encrypted.,2008.

The protection of multimedia data is becoming very important. The protection of this multimedia data can be done with encryption or data hiding algorithms. To decrease the transmission time, the data compression is necessary. There is a new problem trying to combine in a single step, compression, encryption and data hiding. So solutions have been proposed to combine image encryption and compression. A new challenge consists to embed data in encrypted images. Since the entropy of encrypted image is maximal, the embedding step, considered like noise, is not possible by using standard data hiding algorithms. A new idea is to apply reversible data hiding algorithms on encrypted images by wishing to remove the embedded data before the image decryption. Recent reversible data hiding methods have been proposed with high capacity, but these methods are not applicable on encrypted images. Image is encrypted using AES algorithm. AES algorithm consist of four steps which are Substitution Byte, Shift Rows, Mixed columns, Add Round Key. In the encrypted image data is hidden using data hiding algorithm and a secret key. It is used as a seed for PRNG which is used for a substitution-based data hiding technique. Decryption consist of two steps data extraction and image decryption. Data can be extracted using the secret key used for encryption and image is decrypted by using AES algorithm.

2. Lossy compression and iterative reconstruction for encrypted image,2011.

Compression of encrypted data has gained a lot of academic attention in recent years. The usual method for safely and effectively transferring redundant data is to compress it first and then encrypted before transfer. The decryption and decompression procedures are carried out in a logical order at the receiver to retrieve the original data. In lossy compression, the original data is encrypted using a pseudo random string and then compressed with techniques in regard to LDPC.

3. Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption,2013.

This study suggests a method for separably reversibly concealing data in an encrypted grayscale image. The content owner encrypts the image during the first stage by rearranging the pixels using the encryption key. Then, using a data concealing key and a histogram modification method, the data hider inserts some data into the encrypted image. If the receiver merely has access to the encryption key, he can create a copy of the original image but cannot read the hidden data. This image has a Peak Signal to Noise Ratio (PSNR) of 6, which is far greater than what is currently possible. The receiver can extract the data but cannot view the contents of the image if he only has the data hiding key. If the recipient possesses both keys, he may first decrypt the image using the encryption key before extracting the data using the data hiding key. The technique also has a greater capacity for data concealment than the current reversible strategies for data hiding in an encrypted image

4. Encryption of Gray Images using Scalable Codes,2015.

In this study, we used a technique for scalable coding photo encryption. The original pixel values are encrypted using a pseudorandom integer generated from a secret key. Using a multiple-resolution architecture, encrypted data is split into a down sampled sub image and several data sets in the encoding block. In order to reduce the amount of data, an encoder quantizes the sub image and determines the Hadamard coefficients for each data set. Following that, the Hadamard coefficients and quantized sub image are handled as a set of bitstreams. The quantized coefficients can be decrypted at the receiver end using an iteratively updated technique to reconstitute the detailed content, however decrypting the sub image will only yield rough information about the original content. When more bitstreams are received, images of increasing resolution will first be produced after we obtain a lossy rendition of the original image at the output. The PSNR of the reconstructed images is calculated for all scales. When more decomposition levels are used to decrypt encrypted data, PSNR performance increases. In recent years, processing encrypted signals has become a novel and challenging research problem.

5. Reversible Data Hiding,2016.

Data hiding is the practise of concealing data (representing some information) within cover media. In other words, the process of data concealment links the two types of data embedded data and cover media data. The relationship between these two data sources defines many applications. For instance, in clandestine communications, the concealed information might not matter to the cover media. However, the embedded data and the cover material are intimately intertwined when it comes to authentication. In both of these types of applications, buried data must be invisible. The cover media will typically get warped as a result of data concealing in most cases, and it won't be possible to reverse it back to its original condition.

6. Reversible Data Hiding in Encrypted Images with Two-MSB Prediction.2018

In this research, a two-MSB prediction-based reversible data hiding technique for encrypted images is proposed. The suggested method outperforms state-of-the-art techniques in terms of total reversibility, increased embedding efficiency, improved visual quality of the directly decrypted image, and separability between data extraction and original image recovery. The first step is to examine the original image content and identify all potential prediction errors. Next, the original image material is encrypted using the bitwise exclusive or (XOR) operation, and the prediction errors are highlighted based on the binary map of the error locations. Finally, two-MSB substitution is used to incorporate more data into every available pixel. True reversibility, separability, and errorfree data extraction are all achieved by this procedure. The content owner analyses the original image content to detect all possible prediction errors and obtains an error location binary map. Then, the content owner encrypts the original image into an encrypted image by an encryption key Ke using a stream cipher and highlights prediction error location on the encrypted image. In data hiding phase, the data hider without knowing the actual contents of the original images can embed additional data into the processed encrypted image using a data hiding key Kw. For the recipient, if he/she has just the encryption key, only the original image can be recovered perfectly.

7.Authentication by encrypted negative password, 2018.

The input password is hashed into 256 bits value and complement is taken. After complement series of series of permutation and combination and inverse permutation takes place. The obtained new value is called negative password and it is encrypted to form encrypted negative password.

IMPLEMENTATION

1. System Requirements and specification.

The System Requirements Specification document lays out all of the data, functional, and behavioural requirements for the software in development or production. The functionality of a system or one of its subsystems is defined in a functional requirement document. It also relies on the type of program, the number of expected users, and the machine on which the software is installed. A non-functional requirement is one that sets criteria rather than specific behaviour that can be used to judge the operation of a system.

2. Functional requirements.

2.1 Symmetric algorithm.

Symmetric encryption is a type of encryption in which electronic data is encrypted and decrypted using only one key (a secret key). The key must be exchanged between the organizations communicating using symmetric encryption so that it may be utilized in the decryption process. This encryption method varies from asymmetric encryption, which encrypts and decrypts messages using a pair of keys, one public and one private. Data is changed to a form that cannot be understood by anybody who does not have the secret key to decrypt it using symmetric encryption methods. Once the message has been delivered to the intended recipient who holds the key, the algorithm reverses its actions, returning the message to its original and comprehensible state. The secret key used by both the sender and the recipient could be a specific password/code or a random string of letters or numbers generated by a secure random number generator (RNG). The symmetric keys must be generated using a RNG that is certified according to industry standards, such as FIPS 140-2, for banking-grade encryption. There are two types of symmetric encryption algorithms

2.2 Technical Requirements of The System

Hardware Requirements

- System Processor: Core i3
- Hard Disk: 500 GB
- Ram: 4 GB

Software Requirements

- Operating system: Windows 8 / 10

➢ Programming Language: PHP

➢ Software Package: XAMP

SYSTEM ARCHITECTURE

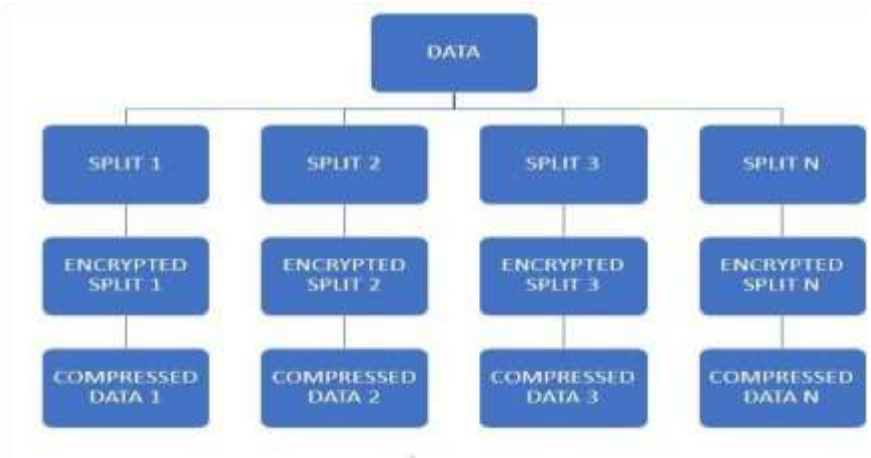


Figure 1. Block diagram of the system.

ALGORITHMS

AES Algorithm

A hash value is appended to each file to identify the sequence of split when a data file is split using a slicer and divided into numerous files. The AES-256 algorithm is used to encrypt each file. Each file is compressed using the LSW technique after encryption. Using the LSW technique, a compressed file is decompressed to reveal an encrypted file. The encrypted file is decrypted using AES-256. The hash value is calculated to determine the split order. The files are organised and concatenated when the order split has been determined.

Each Round's Steps Are:

The algorithm has four steps in each round.

1.Byte replacement: In the first phase, the block text's bytes are substituted according to rules set forth by predefined S-boxes (also known as substitution boxes).

2.Row shifting: The permutation stage is the next. All rows other than the first are relocated by one in this phase, as displayed below. 3.Mixing the columns: The third stage involves using the Hill cypher to further muddle the message by mixing the columns of the block.

4.Adding the round key: The message is XORed with the appropriate round key as the last step. These actions are repeated to guarantee the security of the final ciphertext.

LZW Algorithm

1. Initialization: Fill a dictionary with all single-character codes that could possibly exist. Input data characters are given individual codes.
2. Read Input: Read each symbol of the input data individually.
3. Pattern Recognition and Encoding: To create a series, begin with the initial symbol and gradually add succeeding symbols. Verify the dictionary to see if the sequence is there.

- If the sequence is listed in the dictionary, carry on by adding the following symbol, then repeat this process.
- If the sequence isn't already in the dictionary, display the code for the one that was and then add the new sequence with a new code.

4. Repeat Step 3: Once all of the symbols in the input data have been analysed, carry out the pattern recognition and encoding process again with the remaining symbols.

5.Output: The codes created during the encoding procedure should be output. These codes represent the input data's compressed form.

SHA-256 Algorithm

1.Data Padding: To ensure that the input data has the correct block size for processing, padding is applied. Padding makes ensuring that the input length (512 bits for SHA-256) is a multiple of the block size.

2.Initialization: The algorithm initialises the initial hash state or initial hash digest, which is a set of constant initial hash values. These parameters are present and exclusive to SHA-256.

3.Blocks of defined sizes are created from the padded input data for the message digest computation. Each block in SHA-256 is 512 bits long.

4.Compression Function: The compression function modifies the hash state while working on each block. It creates an updated hash value by combining the most recent block with the most recent hash state.

For each block of the input data, steps 3 and 4 are performed, progressively updating the hash state.

5.a. Finalisation: A finalisation step is carried out to generate the hash output after each block has been processed.

If further padding is required to correspond with the block size, it is added to the final block.

b. Length Append: To maintain data integrity, the length of the initial input data is appended to the last block.

c. Hash Calculation: To generate the final hash value, the compression algorithm is applied to the last block.

6.Output: The SHA-256 digest of the input data is the final hash result, which is commonly shown as a 256-bit (32-byte) hexadecimal string.

Elliptic Curve Cryptography

Key generation

1.A specific elliptic curve and its properties, such as the curve equation, prime modulus, base point, and order, to be chosen.

2. Private Key Generation: Choose a secret private key (d), which is an arbitrary number.

3. Calculate the public key (Q) by multiplying the private key's value by the elliptic curve's base point in a scalar fashion: Q = d * G. On the elliptic curve, the base point (G) is a fixed point with a known order.

Encryption

1. Message Representation: The message to be encrypted should be represented as a point on the elliptic curve. 2. Random Number Generation: Produce a random number (k) from a set of possibilities.

Calculate a point by adding the base point (G) and the message point (M) to get the point (C) on the elliptic curve: C = k * G + M.

The x-coordinate of point C, or a combination of the x-coordinate and other relevant information, is the encrypted message.

Decryption:

1.Using their private key (d), the recipient of the encrypted communication calculates the shared secret. S = d * C. Calculating the Inverse Point: By negating the y-coordinate of the point C, find the inverse point (-C). 2.Decryption: To get the original message point, M = S + (-C), subtract the inverse point (-C) from the shared secret (S).

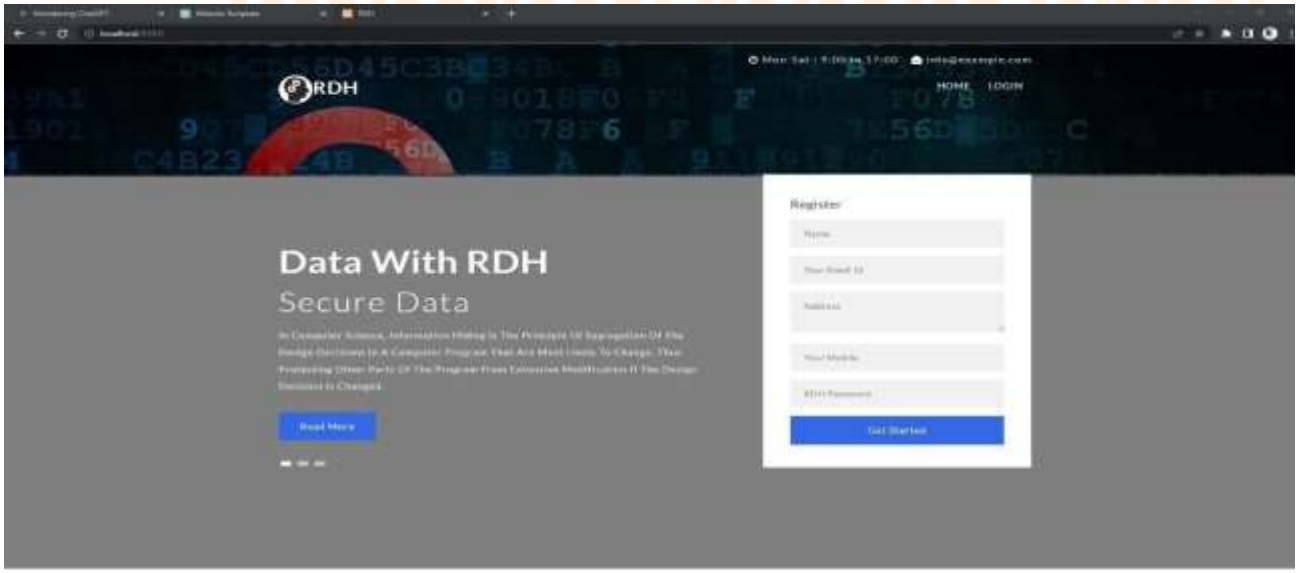
Data Decryption

The encrypted data is ultimately decoded using a decryption algorithm and decryption key. Because a symmetric technique is being used in this project, both the encryption and decryption keys are identical. The receiver may use the original image, which is the decrypted data, for other reasons restores the original state of the encrypted file. Using the same key variable, the data is decrypted. As a result, less computational and processing capacity is available.

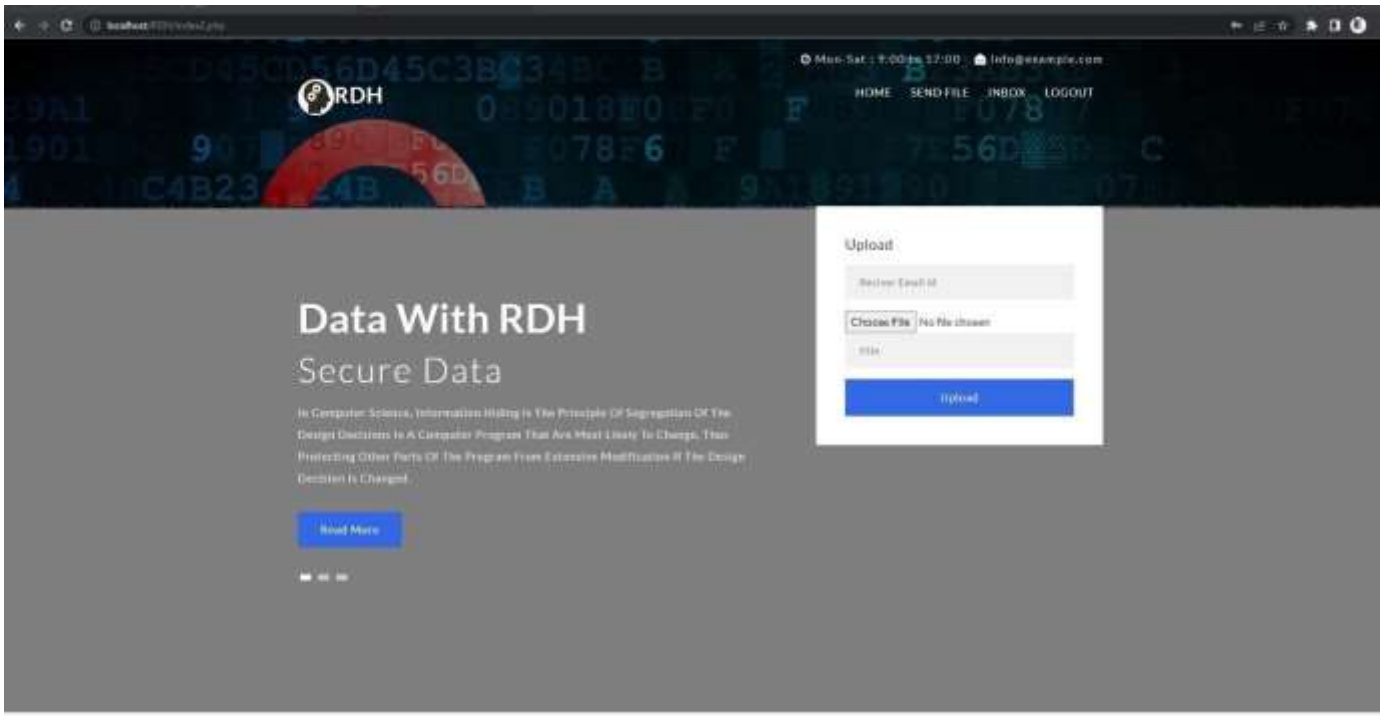
MODULES

Login Page

Registration Page



User Interface



RESULT

We all know that digital communication is becoming increasingly vital in our daily lives. As a result, the communication must be kept private. Therefore, data transit between sender and receiver must be kept secret to prevent unauthorized access. For further protection, the data in this project is split using a slicer then encrypted using encrypted algorithm and compressed using a compression algorithm. The split files are sent to receiver when request is accepted. Files are decompressed and decrypted and combined to get the original data.

REFERENCES

[1] "A Reversible Data Hiding Method Encrypted"., By W. Puech, M. Chaumont and O. Strauss, 2008 [2]

"Lossy compression and iterative reconstruction for encrypted image" By X. Zhang, 2011.

[3] "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption" By Kede Ma, Weiming Zhang, Xianfeng Zhao, Member, 2013.

[4] "Encryption of Gray Images using Scalable Codes" By Guoteng Zhao, Tongqing Wang, Junyong Ye,2015.

[5] "Reversible Data Hiding" By Zhicheng Ni, Yun-Qing Shi, Nirwan, Ansari, and Wei Su. 2016.

[6] "Reversible Data Hiding in Encrypted Images with Two-MSB Prediction" By Yi Puyang, Zhaoxia, Yin, Zhenxing Qian 2018. [7] "Authentication by Encrypted Negative Password ", Wenjian Luo, Yamin Hu, Hao Jiang, and Junteng Wang 2018.